

REMARKS

Applicant wishes to thank the Examiner for the courtesies extended in holding the telephone interview on February 22, 2007. Applicant respectfully submits that the amendments and remarks herein are believed to be consistent with the principles of that discussion.

With this paper, claims 1-3, 5, and 7-30 are presently pending, of which claims 1, 11, 14, and 25 are independent, and claims 27-30 are newly added, while claims 4 and 6 are cancelled without prejudice. Claims 1, 5, 7-9, 11-14, 19, and 24-26 are amended herewith.

The most recent Office Action mailed December 15, 2006 ("*Office Action*") rejected claims 13 and 24 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. The *Office Action* also rejected claims 1-26 under 35 U.S.C. § 112 ¶ 1 as containing subject matter not originally described in the specification at the time the application was filed. Specifically, the *Office Action* noted that the phrase "prior to any user selection . . ." suggested the presence of a negative limitation not otherwise found in the originally-filed application. In light of these rejections under § 101 and § 112, the *Office Action* did not address any of the claims on the merits. In the most recent telephone interview, however, the Examiner seemed to agree that previously presented claim 26 may represent allowable subject matter subject to addressing the § 101 and § 112 rejections of record.

With respect to the § 101 rejection(s) of record, Applicant has amended the claim language in claims 13 and 24 to recite a "computer readable *storage* medium," thereby further distinguishing (in addition to the term "tangible") this limitation from potential claims covering "a modulated signal or carrier wave." As noted for example in ¶ 0020 of Applicant published application, computer readable storage media includes optical or magnetic storage media, such as hard drives, flash media, CD-ROM, DVDs, etc. Applicant respectfully submits, therefore, that the § 101 rejection(s) of record is/are now moot.

Regarding the § 112 rejections, and as discussed during the telephone interview, Applicant herewith amends independent claims 1, 11, 14, and 25 to clarify or remove or qualify the language "prior

to receiving any user selection" (or the like) and respectfully traverses the rejection, as discussed more fully below.

For example, Applicant's invention as recited in amended independent claim 1 includes a system configured to dynamically determine from a database and provide one or more input methods, such as one or more software-based keyboards or input panels to facilitate user input to a plurality of application programs, comprising a plurality of software input methods that are independent of each of the plurality of application programs, each software input method having an input panel configured to receive user input based on user interaction therewith and stored in a software input method database; a software input method manager independent of each of the plurality of application programs, the software input method manager configured in conjunction with an application state determination mechanism to perform the acts of receiving state information from an application program, the state information corresponding to one or more fields that have initially received focus; after receiving the state information from the application, and before receiving any user input in any of the initially focused one or more fields, predicting and selecting an appropriate input panel based on the received state information from the application program; and after receiving initial user input into the initially focused one or more fields, automatically changing the selected input panel based on a subsequent determination of the application program's state.

In addition, Applicant's invention as recited in amended independent claim 11 includes a computer-implemented method for dynamically determining one or more software-based input panels prior to any user selections after opening a particular application program, comprising receiving, from one or more application programs, application state data corresponding to one or more field identifiers of one or more fields that are in focus, the received state data received at a

software input method manager via an application state determination mechanism that is independent of the one or more application programs and independent of the software input method manager, the software input method manager further being independent of the one or more application programs; after receiving the application state data corresponding to one or more field identifiers, and prior to any user selection of a key for input into the one or more fields that are in focus, automatically determining an input panel from a database of input methods for the application program from a plurality of software input methods, each software input method being independent of the application program, wherein the determined input panel is configured for use by the user with the one or more application programs; and returning data to at least one application program corresponding to user interaction with at least one input panel, the at least one input panel having at least one customized, displayed key that, when actuated, returns the text displayed on the key to at least one application program.

Furthermore, Applicant's invention as recited in amended independent claim 14 includes a computer-implemented method for dynamically determining one or more software-based input panels prior to any user selections after opening a particular application program, comprising (i) receiving application program state data from a first application program and a second application program of a plurality of application programs corresponding to one or more initially focused fields, each application program state received at a software input method manager via an application state determination mechanism that is independent of the plurality of application programs and independent of the software input method manager, the application state determination mechanism and the software input method manager independent of the application program corresponding to the application program state data; (ii) upon receiving the application program state data, and prior to any user input into any of the one or more initially focused

fields, selecting one or more input panels from a database of input panels based on the application program state data of the first and second application programs corresponding to the one or more initially focused fields, the input panel being independent of each of the plurality of application programs; (iii) displaying keys on the input panel to enable user interaction with the input panel; and (iv) returning key data to the application program corresponding to user interaction with the input panel.

Still further, Applicant's invention as recited in amended independent claim 25 includes, in a computerized environment comprising a mobile computing device and one or more computerized instructions stored therein that cause the display of an application program and a touch-based input panel at the mobile computing device, a method of, independent of any user selections after opening the application program, automatically determining and displaying one or more customized, touch-based keyboards that are appropriate for a given application program, comprising receiving one or more requests to open an application program; receiving, at a software input method manager, state data from the application program corresponding to one or more fields that have initially received focus; after receiving the one or more requests and the state data, and prior to receiving any user input into the one or more fields, comparing at least a portion of the received state data with an input method selection database information stored in the mobile computing device, the input method selection database comprising information regarding commonly-entered user text for the application program, one or more customized keys unique to the application program, and one or more customized key arrangements for the application program; and displaying through the mobile computing device a customized keyboard that is unique to the application program, wherein the customized keyboard includes

one or more customized keys comprising text that, when selected by the user, is displayed on the mobile computing device.

By contrast, the *Sigl* reference teaches that a device can provide a customized keyboard only *after* a user selects a particular "parameter." *E.g.*, col. 4, ll. 28-33; col. 5, ll. 30-31; col. 5, ll. 61-65. For example, *Sigl* teaches that a user selects a particular parameter, such as "current," whereupon the device provides a keyboard that is specifically customized for that parameter. Specifically, the *Sigl* reference teaches that a user selection of the parameter for current might result in only the display of an alphanumeric keypad, whereupon the user further selects the numbers on the alphanumeric keypad. *Id*; *see also*, col. 4, ll. 30-42. The *Sigl* reference states that this concept of responding to a user's selection of a parameter with a customized keyboard is an "important feature." Col. 5, ll. 35-41.

Along similar lines, the *Dutta* reference teaches that the user initially provides input to select whether to use a default keyboard or a customized keyboard. *E.g.*, col. 5, ll. 1-19. *Dutta* also teaches that the user customizes the keyboard. *E.g.*, col. 3, ll. 57-61; col. 4, ll. 3-26. In particular, *Dutta* teaches that the user's selection for a customized keyboard may allow the user to create a new keyboard, or to use a previously-created custom keyboard. *E.g.*, col. 5, ll. 1-19. In all cases, however, *Dutta* teaches that the customized keyboard is provided only after the user provides some form of input through an interface.

Accordingly, both the *Sigl* and *Dutta* references fail to teach or suggest, whether singly or in combination, each of Applicant's claimed limitations. In particular, neither *Sigl* nor *Dutta* teach or suggest that a software input method manager provides to a user a customized keyboard without the user having to first select a "parameter," without the user having to first enter some input value into a field, or without the user having to first select an option for a customized

keyboard. While there is no question that there is typically at least some user input to execute an application and bring it into "focus," Applicant teaches that a customized keyboard can be provided *before* the user's input into the focused application, while *Sigl* and *Dutta* teach that the customized keyboard is only provided *after* the user's input into the focused application.

Support for the sequence claimed by Applicant is found throughout Applicant's specification. For example, Applicant teaches that an application program can provide the basis for a customized keyboard without first requiring input from the user within an particular field. *E.g.*, ¶ 0007. This is because the application program itself has its own preferences and configurations that can be used to determine a customized keyboard. *E.g.*, ¶ 0007. Thus, not only will the application program relate its state information corresponding to a focused field, but the application program will also (or alternatively) relate its "desires" or "wants" regarding a particular type of key or keyboard to the software input method manager. Specifically, Applicant teaches that:

an *application communicates* with a software input method manager to provide the software input method manager with information related to *a desired [i.e., by the application program] input method*.

Id. (Emphasis added). In addition, Applicant teaches that:

the *application can provide* some of the displayed key choices . . . so that the keys can reflect *what the application wants displayed*.

Id. (Emphasis added). Accordingly, these passages clarify that the type of customized keyboard can be *independent* of user input within a particular application display.

In addition, Applicant's specification teaches that the software input method manager can use the state information provided by the application program to determine an appropriate input panel *both before* user input *and after* user input in a focused field (*i.e.* application window). For

example, Applicant teaches that the software input manager can determine an appropriate input panel:

when the field [application window] initially receives focus, *or . . .* when a user has entered a certain character or string.

Id. (Emphasis added). In other words, the software input manager can identify an appropriate input panel when the application program first begins running (*i.e.*, "initially receives focus"), or after receiving some user input regarding parameters or values (*i.e.*, "has entered a certain character or string"). *Id.*

These distinct alternatives are described throughout Applicant's specification. For example, ¶ 0027 states that an application program need only be "running" to have input focus. In particular, ¶ 0027 teaches that "a number of applications 200 may be executable by the computer system, however *one application that is currently running is said to have input focus.*" In other words, the application program does not require a separate input by the user of a parameter in order to be "focused."

Furthermore, while Applicant does teach that a user's input can provide the basis for selection of a customized keyboard, this is simply one alternative implementation. For example, ¶ 0062 indicates that in "an alternative implementation":

the user [can] manually select an input method for relating to a particular application's field, and then specify (via a checkbox or the like) that when this particular field is focused, the system should thereafter default to the currently selected input method. In this manner, a user may override what an application program or the software input method manager 204 *would otherwise select.*

(Emphasis added). *See also* Figure 9, box 908. The corollary, of course, is that the application program or software input method manager will make *its own selection* of a customized keyboard *in the absence of user input.*

This teaching that the application program and software input panel can communicate concertedly to provide a customized keyboard in the absence of user input within a particular field in focus is further emphasized by Applicant's disclosure of "predicting" a particular input panel. For example, ¶ 0009 discloses that the software input method manager can "automatically change its input panel" (*i.e.*, without input or direction from another source) "based on what the user is *likely to need* for a given application's state." (Emphasis added). In addition, ¶ 0064 discloses that when the application program has communicated a state for which no input method exists, the software input method manager can make "a default input method . . . or *a best guess* for the state, such as if some information is known about it." (Emphasis added). In the latter example, the application program might track what input panel the user chooses as an override option, and then later automatically provide the override option the next time the user executes the application program. *Id.*

In sum, therefore, Applicant's specification teaches the possibility of at least two different scenarios in which the software input method manager will display a particular keyboard. *Compare, e.g.*, ¶ 0007, with ¶ 0062. In a first scenario, which is *before* receiving any user input into a particular field (*e.g.*, a displayed window interface), the software input method manager receives state information from the application program and provides a keyboard that is customized for that application program. *E.g.*, ¶¶ 0007-0009, 0051, 0057-0058, and 0064-0066. In a second scenario, which is *after* receiving user input into a particular field that is in focus, the software input method manager can automatically change the displayed keyboard in order to conform more closely to the user's input. *E.g.*, ¶¶ 0039, 0042, and 0062.

While the *Office Action* appears to suggest that the *Sigl* or *Dutta* are related to the second scenario described above, Applicant respectfully submits that the *Sigl* and *Dutta* references fail

to teach or suggest the first scenario, which is that a user can open an application (*i.e.*, placing it in focus), and then be provided a customized keyboard without having to also enter some input into the focused, running application window. Each of Applicant's independent claims have been amended to make this distinction more clear.

For example, claim 1 recites that the software input method manager receives state information corresponding to one or more fields that have "*initially received focus*" (*i.e.*, just started running), then "after receiving the state information," "and before receiving any user input" into the initially focused fields, the software input method manager predicts and selects "an appropriate input panel." To clarify this limitation, however, claim 1 also recites the second scenario, stating that the software input method manager also "automatically" changes the input panel "after receiving initial user input into the initially focused one or more fields." (Emphasis added).

In addition, claim 11 recites that an input panel is "automatically determine[ed]" from a database "after receiving the application state data corresponding to one or more field identifiers, and prior to any user selection of a key for input into one or more fields that are in focus." Furthermore, claim 14 recites that one or more input panels are selected "upon receiving the application program state data" "corresponding to one or more initially focused fields," "and prior to any user input into any of the one or more initially focused fields." Along these lines, claim 25 recites that a "customized keyboard that is unique to the application program" is displayed "*after* receiving the one or more requests and the state data [corresponding to one or more fields that have initially received focus], *and prior* to receiving any user input into the one or more fields."

Applicant respectfully submits, therefore, that amended independent claims 1, 11, 14, and 25 (and the corresponding dependent claims) are allowable for at least these reasons.

In addition to these independent claims, Applicant respectfully submits that currently amended dependent claims 7-8, 12, and 19 provide additional bases of patentability. For example, claims 7-8, 12, and 19 recite that the application program and software input method communicate "one or more key choices" to the software input method. That is, in addition to simply providing state information regarding a focused field (as in claims 1, 11, or 14), claims 7-8, 12, and 19 recite that the application program can also dictate (*e.g.*, without necessarily any user input into a focused field) the types of keys that are to be displayed. These claim amendments are supported at least in part by ¶ 0007 of Applicant's specification. As Applicant can find no similar teaching or suggestion of this limitation in any of the *Sigl* or *Dutta* references, Applicant respectfully submits that claims 7-8, 12, and 19 represent allowable subject matter.

Furthermore, Applicant respectfully submits that amended claims 9, 25-26, and new claims 27-30 provide still additional bases of patentability. For example, claims 9 and 25-30 recite that at least one of the customized keys provided by the software input method manager can include an entire text string that, when selected, is output onto the display screen. Support for these claim limitations can be found throughout Applicant's specification, and in particular in ¶¶ 0006, 0049, 0052, 0054. As with the previously discussed claim amendments, Applicant can find in the *Sigl* or *Dutta* references for these limitations. Accordingly, Applicant respectfully submits that dependent claims 9, and 26-30 present subject matter that is allowable over *Sigl* and/or *Dutta* for at least these reasons.

Applicant respectfully submits, therefore, that the present application is in condition for allowance. In the event that the Examiner finds remaining impediment to a prompt allowance of

this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 23rd day of February, 2007.

Respectfully submitted,

/Michael J. Frodsham/

RICK D. NYDEGGER
Registration No. 28,651
MIKE J. FRODSHAM
Registration No. 48,699
Attorneys for Applicant
Customer No. 47973

MJF:kbl
CRB0000002035V001